AUTOMATIC BRIDGE DECK DAMAGE USING LOW-COST UAV-BASED IMAGES

Linh Truong-Hong[(1)], Siyuan Chen[(1)], Van Loi Cao[(2)], Debra F. Laefer[(3)]

[(1)] School of Civil Engineering, University College Dublin, Dublin, Ireland
[(2)] NCRA Group, University College Dublin, Dublin, Ireland
[(3)] Center for Urban Science & Department of Civil and Urban Engineering, New York University, US
E-mails: linh.truong-hong@ucd.ie; siyuan.chen@ucd.ie; loi.cao@ucdconnect.ie; debra.laefer@nyu.edu

Abstract: Bridge structures are subjected to deterioration due to excessive usage, overloading, aging, and environmental impacts. Use of visual inspection by live, on-site inspectors predominates the requisite inspection of these structures, despite the known disadvantages of subjective results, high costs, and traffic disruptions due lane closures needed for close-range inspection access. Over the last two decades, significant advancements have occurred in the field of a remote sensing for bridge inspection. Prominent amongst these are use of a point cloud based inspection derived from images collected with an unmanned aerial vehicle (UAV). The approach can rapidly acquire surface details and overcome many of the shortcomings of live, visual inspection but further processing has been required. This paper automates that method for damage inspection of bridge decks. To achieve that, this paper first proposes a robust and efficient method to automatically extract a point cloud of a bridge deck through a cell-based region growing segmentation. Next, locations and areas of the patch deterioration are automatically determined by comparing elevations of the point clouds to the surface of the undamaged bridge deck. Finally, a deep learning method, using a one-class autoencoder, is employed to classify the point cloud of the bridge deck into cracking area and undamaged one.

Keywords: bridge inspection, bridge deck, UAV, point cloud, segmentation, patch deterioration, cracking, deep learning, autoencoder

1. Introduction

The most recent American Society of Civil Engineers' (ASCE) Report Card on infrastructure reported more than 50% of that nation's 614,387 bridge are more than 40-years old with about 9.1% of them exhibiting structural deficiencies (ASCE 2017). Similar quantities of aging bridges were also found in a 6 European nation study, where the majority of bridges were built in the period 1945-1965, undergone significant deterioration (Pakrashi, et al. 2011). Assuming a 50-year designed lifespan, the current performance of any of the bridges must be assumed to be significantly reduced due to age coupled with a variety of likely factors including excessive usage, overloading, material aging, and environmental impacts. As such, an accurate assessment of such a bridge's condition is needed for maintaining a safe, functional, and reliable structure. Additionally, inspection is needed to update asset inventories to assist in bridge management. In practice, visual inspection with on-site inspectors associated with special equipment is the predominant method. This approach, albeit the most common one, has many downsides including being subjective and highly dependent upon an inspectors' experience, being slow and expensive inspectors, and requiring traffic delays and closures (Metni and Hamel 2007; Phares et al. 2004).

In contrast, with the development of robotics and computer vision, low-cost UAVs have been introduced and widely used in topographic surveying. The method can capture high resolution images of surfaces from an air, and then high dense, accurate three-dimensional (3D) data points of the components can be generated. From such point clouds, features and 3D models of objects can be extracted. That argues for use of UAV for bridge inspection as an alternative or complimentary method, as it does not require traffic closure or experienced, on-site inspectors.

Initial UAV deployments with imagery capabilities have focused on specific tasks. For example, Kim et al. (2015) aimed to detect cracks in bridge superstructures. Ellenberg et al. (2016) analysed an error budget of an UAV with an integrated 10-megapixel camera measuring in situ displacements of a steel girder in a laboratory. Result showed deviations of up to 31.5%. Recent efforts on reconstructing 3D point clouds from overlapping images have been developed for bridge assessment, but reported accuracies are relatively low. For example, Neitzel and Klonowski (2011) showed point cloud-based images having an absolute deviation of up to 20 cm. Moreover, Escobar-Wolf et al. (2017) used infrared and digital single-lens reflex (DSLR) Nikon D800 cameras mounted on UAV to image the bridge deck to identify delamination. The study involved the Merriman and Stark Road overpass bridges, on highway I-96 in Detroit, Michigan, and reported that infrared and DSLR images can give acceptable results of delamination. Recently, Lovelace (2015)

investigated UAV technology for bridge safety inspections. The focus was capturing the surface of major bridge components of 4 bridges in Minnesota and concluded that while UAVs can aid in bridge inspection and that defects from UAV-based images are detectable, the current generation of hardware and supporting imagery software cannot replace on-site inspectors where tactile functions (cleaning or measuring) are needed to detect damage or see its full extent.

2. A proposed method

In light of these advancements, this paper proposes a robust and efficient method to automatically extract a point cloud of a bridge deck generated from a bundle image captured by a low-cost UAV (Fig. 1). The proposed workflow consists of 3 main parts: (I) data acquisition and preprocessing, (II) bridge deck extraction and (III) Bridge deck damage detection. Part II involves decomposing the point cloud into 2D adjoining cells and using a cell-based region growing approach to group adjacent cells having deviations features satisfying predefined conditions. Then, backward and forward algorithms were applied to minimize over- and/or under-segmentation. In Part III, locations and areas of patch deterioration are determined by comparing selected (i.e. damaged) sections to non-selected (i.e. undamaged) section of the bridge deck's surface. Finally, in Part III, a deep learning method, an autoencoder-based one class, was proposed to extract cracking location of the bridge deck.

This paper mainly focuses on Parts II and III, as Part I employs commercial software to generate a point cloud images captured from the UAV captured images and uses an open-source software for semi-automatic denoising. To facilitate identifying bridge pavement surface damage, in Part II, a robust, efficient segmentation is proposed to automatically extract the point cloud of the bridge pavement surface. Part III ultimately reports the condition of the bridge deck based on patch deterioration and cracking.

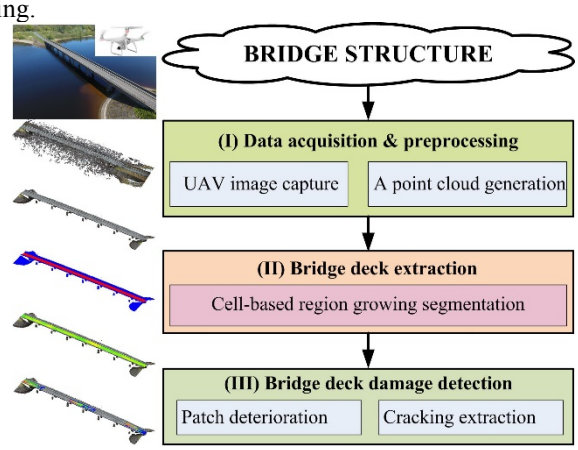


Fig. 1. Proposed workflow

3. Test-bed bridge

To test the proposed workflow, the Blessington bridge connecting Kilbride Rd. to Lake Dr Rd., crossing Liffey Lake in County Wicklow, Ireland was selected. The bridge is of reinforced concrete, about 130 m long and 8m wide with 2 traffic lanes plus 1 pedestrian lane (Fig. 2). A low-cost UAV, DJI Phantom 4 quadrotor with a 4K camera was used to capture the bridge. The 4K camera has a field of view of 94°, and a focal length of 35 mm. This can take an image a 4000 x 3000 pixel image, which translate to a ground sampling distance of 10 mm/pixel from a height of 15 m.

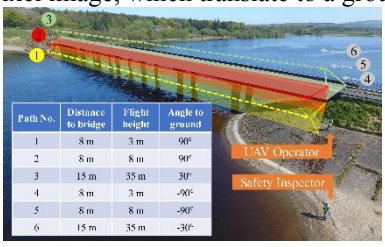


Fig. 2. Blessington bridge and flight paths for data capture

a) A point cloud generated from images b) A point cloud after removing noise data

Fig. 3. A point cloud of the bridge from a image acquired from UAV

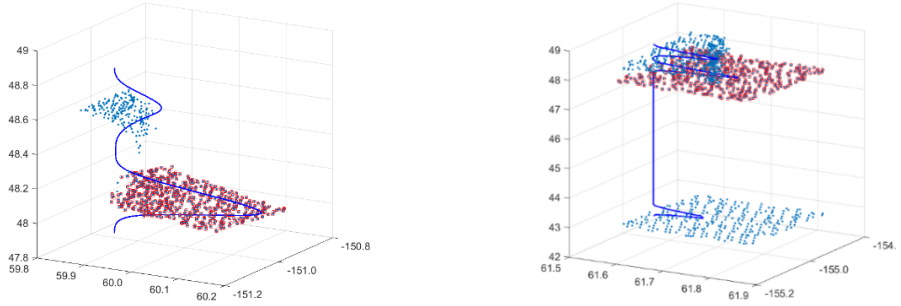
The data capture strategy was designed to ensure comprehensive coverage of the bridge deck. This involved 7 parallel flight paths (2 along each side, 3 above the desk) at 8 m -15 m above the bridge deck. A total of 212 images were collected within the 1 hour of flying. These were converted into 3D data point cloud with Agisoft PhotoScan (AgiSoft 2017), in Part I of the procedure. The software eliminates distorted or blurred images and those with inadequate overlap to other images (Siebert and Teizer 2014). Based on the automatic detection of common features, the images are aligned by the software to form a single point cloud, which in this case involved more than 16.8 million points [each with an x, y, z coordinate and associated red-green-blue (RGB) colors (Fig. 3a)]. The process took about 1 hour on a Dell XPS with i7

CPU with a speed clock 2.8 GHz, 4 cores and 16 Gb RAM, on Window 10 operation system. Since, the presence of the water's waves and shadowing beneath the bridge complicates the data set, the next step involved eliminating these points, which can be considered as noise. This was done in CloudCompare V2.9 (Filter 2018), an open-source software. After noise removal (Fig. 3b), approximately 12.5 million points remained for further processing; more details of the noise removal can refer to Chen et al. (Chen, et al. 2018).

4. Bridge deck extraction

A goal of Part II is to extract the bridge deck from within the point cloud. A cell-based region growing was adapted for this step. The process begins by decamping the 3D point cloud into a 2D horizontal plane and a 2D bounding box demarcating the boundaries of entire the input data. The bounding box is defined as a pair of the corners [x_{min}, y_{min}] and [x_{max}, y_{max}] derived from the 3D point cloud. Next, a quadtree (Samet 1984) is employed to recursively subdivide the bounding box into 4 smaller adjoining 2D cells until the termination criteria, which in this case was only a maximum cell size of 0.5 m. Cells were classified as "*empty*" if the cell contained no data points or "*full*" if at least one point was present.

As a full cell may contain point clouds from multiple bridge components (e.g. safety rail, bridge deck and pier, as shown in Fig. 4), a new algorithm was devised to eliminate bridge deck points. By observing a distribution of the point cloud within the cell in a vertical direction, it can be seen the point cloud of the bridge deck often in a form of the highest peak of a probability density shape. For example, the blue points in Fig. 4 showed the point cloud in the cells, while the red circle filled blue points are the point cloud of the bridge deck. Results of this filtering are to secure the cells only contains a point cloud of one of bridge's component.



a) Point clouds of safety rail and bridge deck b) Point clouds of curb, bridge deck and pier

Fig. 4. Cells containing point clouds of different bridge components

Next, a cell-based region growing is to partition a 3D point cloud of a bridge into sub-set point clouds representing to individual surfaces. In this study, it is assumed that the point cloud within the cell representing to a patch of the surface. As such, local surface features of each cell, which includes its normal vector and the residual, are computed for a segmentation process. The points within each cell are assumed to describe a plane for which the centroid (p_0) of the surface (S) can be expressed as Eq 1. Using principal component analysis (PCA) enables determination of the surface normal n = (n_x, n_y, n_z), which is the eigen-vector corresponding to the smallest eigen-value determined from the covariance matrix given in Eq 2. The surface of the cell can be expressed in the form S (p_0, n). Finally, the distances, d(**p**, S), from all points in the cell to the surface S are computed, and the residual (res) is defined as the root mean square of these distances.

$$p_0 = \frac{1}{N}\sum_{i=1}^{N} p_i^r \quad (1)$$

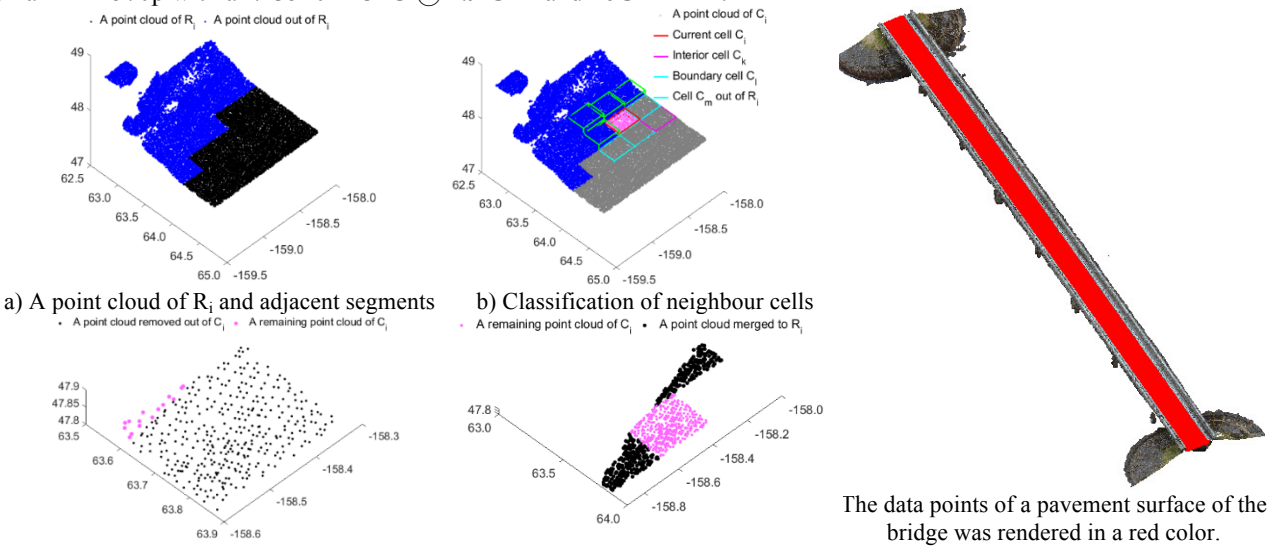
$$C = \sum_{i=1}^{N}(p_i^r - p_0)(p_i^r - p_0)^T \quad (2)$$

where **p** = (x_i, y_i, z_i) ∈ R^3 is x-, y-, z- coordinates of the data points.

The cell-based region growing is a process to incrementally group adjoining cells having the local surface features satisfying conditions. The process starts with the cell having the smallest residual, which is herein called the searching cell. Adjoining cells are added to the region of the searching cell, if difference of the normal vectors and elevations is less than 2.5°and 50 mm respectively, as established empirically. Having two conditions help ensure continuity of the region. New adding cells are then considered searching cells, if the residual values of these cells are less than the residual threshold, which is empirically selected by 10 mm. This condition aims to set the seeding cell must be represented a smooth patch of the surface. The process continues until all cells are exhausted. The process implemented herein is similar to works of Vo et al. (Vo et al. 2015).

Since a cell on a region boundary may contain multiple component surfaces (e.g. a traffic lane and a pedestrian path), results at this stage may be over- and/or under-segmented. To address this, backward and forward algorithms are introduced. The backward algorithm is to solve the over-segmentation issue, where the point clouds of other regions were

segmented to the current region. The forward algorithm is for the under-segmentation, where the point clouds of the current region have not yet included can be seek and merged to the region. From the cell C_i on the boundary of region R_i, its neighbour cells, $\mathbf{C}_j$ were extracted. The C_j points were classified into 1 of 3 groups: (1) interior cell(s), C_k of R_i, (2) boundary cell(s), C_l of R_i, and (3), out-region cell(s) C_m of R_i. Next, the fitting surface S_l was created from the point clouds within C_k by using PCA. In the backward algorithm, the points (p_i) in C_i were removed, if the distance, d(**p**, S_l), exceeds than the distance threshold of 10 mm. Similarly, in the forward algorithm, points (**q**) in C_m were merged into R_i if the distance, d(**q**, S_l), fails to exceed the distance threshold of 10 mm. Figure 5a-d illustrates the backward and forward algorithms, while Fig. 5e shows the automatically extracted bridge deck. Finally, performance pf the proposed method was evaluated by comparing the point cloud of the bridge pavement extracted to manually extracted reference data. This resulted in an F1-score of 0.95, with an execution time of only 171.4 seconds when implemented in Matlab script and run on an HP 2570p with a i7-3520M CPU @ 2.9 GHz and 16GB RAM.



a) A point cloud of R_i and adjacent segments b) Classification of neighbour cells c) Result of the backward algorithm d) Result of the forward algorithm e) Segmentation results

Fig. 5. Cell-based region growing segmentation for extracting a pavement surface of a bridge

5. Bridge deck condition

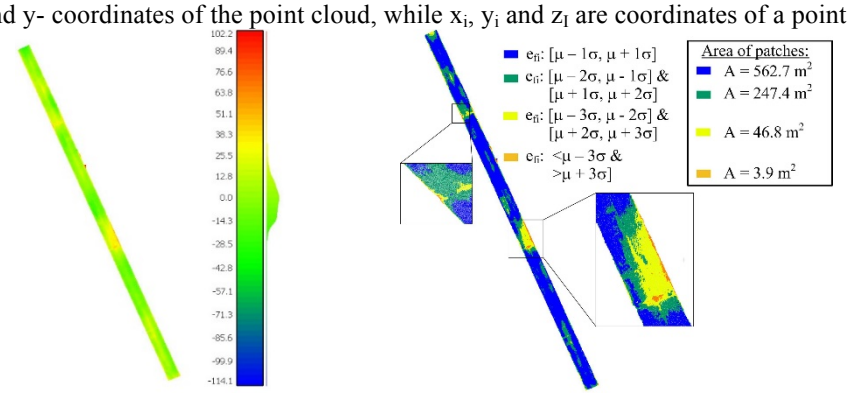
5.1 Patch deterioration

A goal of Part III is to identify bridge deck damage in the form of patch deterioration and cracking, although other damage types may be present. The patch deterioration is parts of the surface to be loosed and/or filled additional material during road pavement maintenance. In this case, patching often differs from undamaged areas in term of geometry and possibly texture. As such, in this study, areas were considered as the patch deterioration if the elevations of the point clouds of the areas differ from the un-damaged surface, which can be obtained by fitting through all point clouds of the bridge deck. However, to minimize negative impact of the point cloud of damaged area to the fitting surface, it was assumed the patch deterioration was taken account of a small portion of the bridge deck. The fitting surface of the bridge deck (expressed in Eq 3) can be obtained by employing a 3D curve fitting algorithm the built-in Matlab software (MathWorks 2016). The fitting results showed an R-square value of 0.998 with an RMS of 12.7 mm, implying that the fitting surface can be represent the undamaged condition of the bridge deck. Additionally, elevation differences (e_f) from the point cloud to the fitting surface were then computed according to Eq 4.

$$S(x, y) = 47.32 - 0.03*x + -0.04*y - 0.002xy - 0.003*x^2 - 0.0007*y^2 \quad (3)$$

$$e_{fi} = z_i - S(x_i, y_i) \quad (4)$$

where x and y are x- and y- coordinates of the point cloud, while x_i, y_i and z_I are coordinates of a point p_i in the data set.



a) Distribution of e_f b) Distribution of patch deterioration

Figure 6. Identifying patch deterioration of a bridge deck

Results showed e_f varying from -114.1 mm to 100.2 mm, with a mean (μ) of 0.0 mm and the standard deviation (σ) of 12.7 mm (Fig. 6a). The area of each range e_f is shown in Fig. 6b. Patch deterioration with e_f in the range from $\mu \pm 1\sigma$ to $\mu \pm 2\sigma$ is around 247.8 m^2 with 28.8% of the bridge deck (Fig. 6b). Moreover, the damaged patch had an e_f 25.4mm - 38.1mm ($[\mu + 2\sigma, \mu \pm 3\sigma]$) across 46.8 m^2 or about 5.4% of the total area of the bridge deck. Interestingly, damage patches having an e_f larger than 38.1mm are mostly on the boundary of the pavement surface. That is because the over-segmentation still presents, where parts of the point cloud of the pedestrian lane were included in the bridge deck segment.

5.2 Cracking extraction

In undamaged areas, a point cloud's sampling step (i.e. distance between two adjacent data points) is typically equal, except for missing data. The sampling step is larger at locations where cracking is present because of surface discontinuities in these locations. According to Laefer et al. (2014) a crack can be detected in a point cloud if the crack width is larger than two times the sampling step. However, in practice, noise and mixed data points interfere with this process (Truong-Hong et al. 2016). Interestingly, by observing the texturing surface, it can be seen that the cracks present as dark regions while un-cracked areas are bright ones (Valença et al. 2017). Therefore, colours of the point cloud can be good features for classifying whether the points in the crack area or not.

Based on these observations, a machine learning technique in the form of Autoencoders-based one class was employed to identify cracked regions on the bridge pavement. An autoencoder is a neural network consisting of two parts: *encoder* and *decoder*. The *encoder* plays as a feature extractor that explicitly represents input data X in a feature space. Let f_θ denote the *encoder*, and $X = \{x_1, x_2, ..., x_n\}$ be a dataset. The *encoder* f_θ maps the input x_i into a latent vector $z_i = f_\theta(x_i)$, where z_i is the *code* or *latent representation* of x_i. The *decoder* g_θ map z_i back into the input space, which forms a reconstruction $\hat{x}_i = g_\theta(z_i)$. The *encoder* and *decoder* are commonly represented as single-layer neural networks in the form of non-linear functions of affine mappings as follows:

$$f_\theta = \psi_s(W_1 x + b_1) \quad (5)$$

$$g_\theta = \psi_g(W_2 x + b_2) \quad (6)$$

where W_1, W_2 and b_1, b_2 are the weight matrices and biases of the *encoder* and *decoder*, ψ_s and ψ_g are the activation functions of the *encoder* and *decoder*, such as a *sigmoid* or *tanh* function.

Autoencoders learn to minimize the reconstruction error (RE) between the input and its reconstruction values at the output layer with regard to the parameters $\theta = \{W_1, W_2, b_1, b_2\}$. The reconstruction loss function can be the *mean squared error* (MSE) or the *cross-entropy* loss. By compressing input data into a lower dimensional space, the classical autoencoder avoids simply learning the identity and removes redundant information (Japkowicz, et al. 1995). As such, AEs are often applied for constructing one-class classifiers (Cao et al. 2016; Cao et al. 2018). Thus, the REs can be used as classification score, which means that a query point having classification score below a pre-determined threshold is classified in the in-class, and otherwise it is categorized as the out-class.

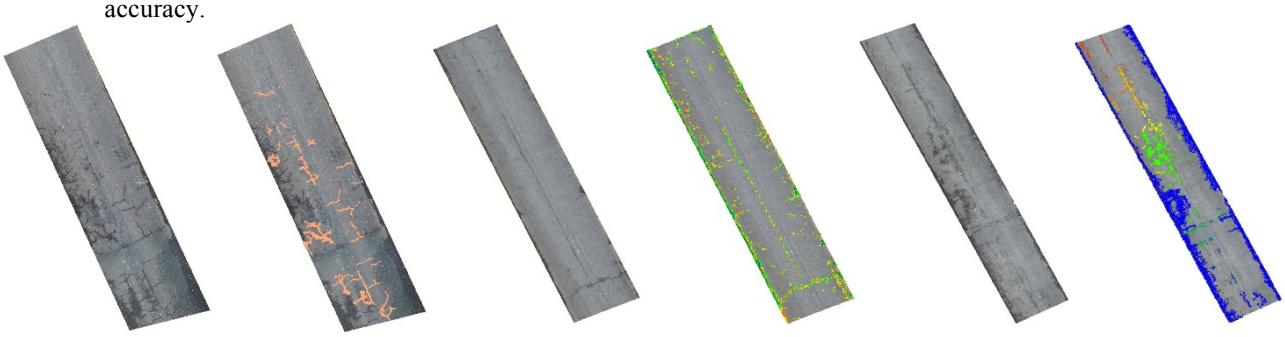
In this experiment to develop a model to predict crack areas, point clouds of the crack regions were manually extracted (Fig. 7a). A total of 12,164 points was used as a training data set. The AE was designed with two hidden layers, and three neurons in each, with the aim to reduce the number of hyper-parameters of the AE, *Adadelta* algorithm (Zeiler 2012) together with early-stopping techniques (Prechelt 1998) were employed.

After a learning model was created, two data sets extracted from the entire data set of the pavement surface of the bridge deck were used to predict crack locations. Results of the prediction are shown in Fig. 7b and c. A visualisation evaluation showed that most of the cracks in Data set 2 and 3 were extracted (Fig. 7b and c). However, the method was given to over-prediction, where along boundaries of the bridge deck, where these regions were mistakenly classified as cracks because of having similar colors to the crack locations. This issue can be eliminated by introducing a further step comparing shapes of the predicted cracking to a real one. Moreover, additional attributes of the point cloud may be included in a training model to improve an accuracy of the classifier.

6. Conclusions

1. A cell-based region growing was proposed to automatically extract a point cloud of the bridge deck. The evaluation was showed the proposed method can extract the bridge deck with F1-score of 0.95 and processed more than 12 million data points in 174.1 seconds
2. Patch deterioration of the bridge deck was automatically identified by comparing elevations of the point cloud to the undamaged bridge deck surface. Both location and area of the patch can be reported and used for decision making and bridge management. However, an evaluation of the proposed method should be evaluated against an independent method.
3. Implementation of deep learning in the form of an Autoencoder-based one class was successful in extracting

most of cracks in the bridge deck, in which only RGB colors of the point cloud were used in the training model. Over-extraction occureed along bridge deck boundaries, but can be solved by analysing shapes of the extracted cracks. Furthermore, additional attributes of the point cloud can be used in learning the model to improve an accuracy.



a) Data set 1 and the training data b) Data set 2 and resulted cracking c) Data set 3 and resulted cracking

Note: color points is either a training data or resulted cracking predicted from a machine learning model

Fig. 7. Illustrated results of cracking extraction from Autoencode-based one class

Acknowledgements

This work was funded with the generous support of the UCD Seed funding for the project Laser Scanning for Automatic Bridge Rating, grant SF1404. The first author is grateful for this support.

References

ASCE, 2017 infrastructure report card: Bridge, Report No.: 2017.

Cao, V.L.; Nicolau, M.; McDermott, J. 2016, A hybrid autoencoder and density estimation model for anomaly detection, in *Proc. of International Conference on Parallel Problem Solving from Nature*, 2016, pp. 717-726

Cao, V.L.; Nicolau, M.; McDermott, J. 2018, Learning neural representations for network anomaly detection, *IEEE Transactions on Cybernetics*, 1-14, 10.1109/TCYB.2018.2838668.

Chen, S.; Truong-Hong, L.; O'Keeffe, E.; Laefer, D.F.; Mangina1, E. (in press 2018), Outlier detection of point clouds generating from low cost uavs for bridge inspection, in *Proc. of The Sixth International Symposium on Life-Cycle Civil Engineering, IALCCE 2018* Ghent, Belgium, 28-31 October 2018

Cloudcompare-noise filter 2018, [GNU GPL], Version 2.9.0, Retrieved from: http://www.cloudcompare.org/

Ellenberg, A.; Kontsos, A.; Moon, F.; Bartoli, I. 2016, Bridge related damage quantification using unmanned aerial vehicle imagery, *Structural Control and Health Monitoring* 23 (9), 1168-1179

Escobar-Wolf, R.; Oommen, T.; Brooks, C.N.; Dobson, R.J.; Ahlborn, T.M. 2017, Unmanned aerial vehicle (uav)-based assessment of concrete bridge deck delamination using thermal and visible camera sensors: A preliminary analysis, *Research in Nondestructive Evaluation*, 1-16

Japkowicz, N.; Myers, C.; Gluck, M. 1995, A novelty detection approach to classification, in *Proc. of IJCAI*, Vol. 1, 1995, pp. 518-

Kim, J.-W.; Kim, S.-B.; Park, J.-C.;Nam, J.-W. 2015, Development of crack detection system with unmanned aerial vehicles and digital image processing, *Advances in structural engineering and mechanics (ASEM15)*, MISSING PAGES?

Laefer, D.F.; Truong-Hong, L.; Carr, H.; Singh, M. 2014, Crack detection limits in unit based masonry with terrestrial laser scanning, *NDT & E International* 62, 66-76

Lovelace, B. 2015, Unmanned aerial vehicle bridge inspection demonstration project, *Minnesota Department of Transportation*, 1-44

Mathworks: Curve fitting toolbox 2016, [Commercial], Version: 2016a, Retrieved from: https://uk.mathworks.com/products/curvefitting.html

Metni, N.; Hamel, T. 2007, A uav for bridge inspection: Visual servoing control law with orientation limits, *Automation in Construction* 17 (1), 3-10

Neitzel, F.; Klonowski, J. 2011, Mobile 3d mapping with a low-cost uav system, *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci* 38, 1-6

Pakrashi, V.; O'Brien, E.;O'Connor, A. 2011, A review of road structure data in six european countries, *Proceedings of the ICE-Urban Design and Planning* 164 (4), 225-232

Phares, B.M.; Washer, G.A.; Rolander, D.D.; Graybeal, B.A.;Moore, M. 2004, Routine highway bridge inspection condition documentation accuracy and reliability, *J Bridge Eng* 9 (4), 403-413

Photoscan pro 2017, [Commercial], Version 1.3.2, Retrieved from: http://www.agisoft.com/downloads/installer/

Prechelt, L., Early stopping-but when?, Neural networks: Tricks of the trade, Springer, 1998, pp. 55-69.

Siebert, S.;Teizer, J. 2014, Mobile 3d mapping for surveying earthwork projects using an unmanned aerial vehicle (uav) system, *Automation in Construction* 41, 1-14, https://doi.org/10.1016/j.autcon.2014.01.004.

Samet, H. 1984, The quadtree and related hierarchical data structures, *ACM Comput. Surv.* 16 (2), 187-260, 10.1145/356924.356930.

Truong-Hong, L.; Falter, H.; Lennon, D.; Laefer, D.F. 2016, Framework for bridge inspection with laser scanning, in *Proc. of EASEC-14 Structural Engineering and Construction* Ho Chi Minh City, Vietnam, 6-8 January 2016

Valença, J.; Puente, I.; Júlio, E.; González-Jorge, H.; Arias-Sánchez, P. 2017, Assessment of cracks on concrete bridges using image processing supported by laser scanning survey, *Construction and Building Materials* 146, 668-678

Vo, A.-V.; Truong-Hong, L.; Laefer, D.F.; Bertolotto, M. 2015, Octree-based region growing for point cloud segmentation, *ISPRS Journal of Photogrammetry and Remote Sensing* 104, 88-100, http://dx.doi.org/10.1016/j.isprsjprs.2015.01.011.

Zeiler, M.D. 2012, Adadelta: An adaptive learning rate method, *arXiv preprint arXiv:1212.5701*